ABSTRACT
This paper describes a new low-complexity full-band (20 kHz) audio coding algorithm which has been recently standardized by ITU-T as Recommendation G.719. The algorithm is designed to provide 20 Hz - 20 kHz audio bandwidth using a 48 kHz sample rate, operating at 32 - 128 kbps. This codec features very high audio quality and low computational complexity and is suitable for use in applications such as videoconferencing, teleconferencing, and streaming audio over the Internet. Subjective test results from the Optimization/Characterization phase of G.719 are also presented in the paper.

Index Terms—Audio coding, full-band, low complexity, adaptive time-frequency transform, fast lattice vector quantization

1. INTRODUCTION
In hands-free videoconferencing and teleconferencing markets, there is strong and increasing demand for audio coding providing the full human auditory bandwidth of 20 Hz to 20 kHz. This is because:

- Conferencing systems are increasingly used for more elaborate presentations, often including music and sound effects (i.e. animal sounds, musical instruments, vehicles or nature sounds, etc.) which occupy a wider audio band than speech. Presentations involve remote education of music, playback of audio and video from DVDs and VCRs, audio/video clips from PCs, and elaborate audio-visual presentations from, for example, PowerPoint.
- Users perceive the bandwidth of 20 Hz to 20 kHz as representing the ultimate goal for audio bandwidth. The resulting market pressures are causing a shift in this direction, now that sufficient IP bitrate and audio coding technology are available to deliver this.

As with any audio codec for hands-free videoconferencing use, the requirements include:

- Low latency (support natural conversation)
- Low complexity (free cycles for video processing and other audio processing; reduce cost)
- High quality on all signal types

To meet the market need for such a full-band audio coding standard, ITU-T Q10/SG16 launched the standardization of Low-Complexity Full-Band Audio Coding Extension to G.722.1 (G.722.1FB) in November 2006. The ITU-T G.722.1FB

Figure 1: Block diagram of the G.719 encoder.
A block diagram of the G.719 encoder is shown in Figure 1. The input signal sampled at 48 kHz is processed through a transient detector. Depending on the detection of a transient, indicated by a flag IsTransient, a high frequency resolution or a low frequency resolution transform is applied on the input signal frame. The adaptive transform is based on a modified discrete cosine transform (MDCT) [2] in case of stationary frames. For transient frames, the MDCT is modified to obtain a higher temporal resolution without a need for additional delay and with very little overhead in complexity. Transient frames have a temporal resolution equivalent to 5 ms frames.

The obtained transform coefficients are grouped into bands of unequal lengths as 8, 16, 24, or 32. As the bandwidth is 20 kHz, only 800 transform coefficients are used. The 160 transform coefficients representing frequencies above 20 kHz are ignored. The norm or power of each band is estimated and the resulting spectral envelope consisting of the norms of all bands is scalar quantized and encoded. The coefficients are then normalized by the quantized norms. The quantized norms are further adjusted based on adaptive spectral weighting and used as input for bit allocation. An adaptive bit-allocation scheme based on the quantized norms of the bands is used to assign the available bits in a frame among the bands. The bit-allocation which is essential for decoding quantization indices for both the coded norms and transform coefficients is estimated, coded, and transmitted to the decoder.

2.2 Decoder

<table>
<thead>
<tr>
<th>FLVQ Decoding</th>
<th>Spectral-fill Generator</th>
<th>DEMUX</th>
<th>Noise level</th>
<th>Norm indices</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transient signaling</td>
<td>FLVQ indices</td>
<td>Norm adjustment index</td>
<td>Envelope shaping</td>
<td>Inverse Transform</td>
</tr>
<tr>
<td>Audio signal Fx = 48 kHz</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 2: Block diagram of the G.719 decoder.

A block diagram of the decoder is shown in Figure 2. The transient flag is first decoded which indicates the frame configuration, i.e. stationary or transient. The spectral envelope is then decoded and the same, bit-exact, norm adjustment and bit-allocation algorithms are used at the decoder to re-compute the bit-allocation which is essential for decoding quantization indices of the normalized transform coefficients.

After decoding the transform coefficients, the non-coded transform coefficients (allocated zero bits) in low frequencies are regenerated by using a spectral-fill codebook built from the decoded transform coefficients. The noise level adjustment index is used to adjust the level of the regenerated coefficients. The non-coded transform coefficients in high frequencies are regenerated using bandwidth extension. The decoded transform coefficients and regenerated transform coefficients are mixed and lead to normalized spectrum. The decoded spectral envelope is applied to the decoded full-band spectrum. Finally, the inverse transform is applied to recover the time-domain decoded signal. This is performed by applying either the inverse MDCT for stationary mode, or the inverse of the higher temporal resolution transform for transient mode.

3. ADAPTIVE TIME-FREQUENCY TRANSFORM

The adaptive time-frequency transform is based on the detection of a transient. In the case of stationary signals, the transform has a high frequency resolution which is able to efficiently represent stationary sounds. In the case of transient sounds, the time-frequency transform will increase its time resolution and allows a better representation of the rapid changes in the input signal characteristics. These two modes of operations share a common buffering and windowing module and the switching between one mode of operation to the other is instantaneous. Thus no additional look-ahead in the transient detection is needed. This allows the codec to have selectable time resolution at low complexity and with zero additional delay.

If the input signal is detected as stationary, a type IV discrete cosine transform (DCT_IV) [2] is applied on the output \( \tilde{x}(n) \) of the time domain aliasing (TDA) operation. The DCT_IV of the time aliased signal is defined by the following equation:

\[
y(k) = \sum_{n=0}^{959} \tilde{x}(n) \cos \left[ \frac{n + \frac{1}{2}}{n} \left( k + \frac{1}{2} \right) \frac{\pi}{960} \right], k = 0, 1, ..., 959 \quad (1)
\]

The resulting signal \( y(k) \) represents the transform coefficients of the input frame. It should be noted that in stationary mode, the cascade of windowing, TDA, and DCT_IV is equivalent to applying the modulated lapped transform (MLT) [2].

If the signal is detected as a transient, a further re-ordering of the time-domain aliased signal is performed. The basis functions of the resulting filter-bank would have an incoherent time and frequency responses without re-ordering. The re-ordering operation consists of shuffling the upper and lower half of the TDA output signal \( \tilde{x}(n) \) as follows:

\[
v(n) = \tilde{x}(959 - n), n = 0, 1, ..., 959. \quad (2)
\]

This re-ordering is only conceptual and in reality no computations are involved. Higher time resolution is obtained by zero padding the signal \( v(n) \) and dividing the resulting signal into four overlapped equal length sub-frames. The amount of zero-paddings is equal to 120 on each side of the signal. The segments are 50% overlapped and each segment has a length equal to 480. The two inner segments are post-windowed using a sine window of length 480. The windows for outer segments are constructed using half a sine window.

Each resulting post-windowed segment is further processed by applying the MDCT, i.e. time aliasing followed by DCT_IV. The output of the MDCT for each segment represents the signal spectrum at different time instants, thus in case of transients, a higher time resolution is used. The length of the output of each of the four MDCT is half the length of the input segment, i.e. 240. Therefore, this operation does not introduce additional redundancy.
4. TRANSFORM COEFFICIENT QUANTIZATION

Each band consists of one or more vectors of 8-dimensional transform coefficients and the coefficients are normalized by the quantized norm. All 8-dimensional vectors belonging to one band are assigned the same number of bits for quantization. A fast lattice vector quantization (FLVQ) scheme [3] is used to quantize the normalized coefficients in 8 dimensions. In FLVQ the quantizer comprises two sub-quantizers: a $D_8$-based higher-rate lattice vector quantizer (HRQ) and an $RE_8$-based lower-rate lattice vector quantizer (LRQ).

HRQ is a multi-rate quantizer designed to quantize the transform coefficients at rates of 2 up to 9 bit/coefficient and its codebook is based on the so-called Voronoi code for the $D_8$ lattice [4]. $D_8$ is a well-known lattice and defined as:

$$D_8 = \{(y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8) \in Z_8 | \sum_{j=1}^{8} y_j = \text{even}\}$$

where $Z_8$ is the lattice which consists of all points with integer coordinates. It can be seen that $D_8$ consists of the points having integer coordinates with an even sum. The codebook of HRQ is constructed from a finite region of the $D_8$ lattice and is not stored in memory. The codewords are generated by a simple algebraic method and a fast quantization algorithm is used.

To minimize the distortion for a given rate, the $D_8$ lattice should be truncated and scaled. Actually, the input vectors instead of the lattice codebook are scaled in order to use the fast searching and indexing algorithms proposed by Conway and Sloane [4]. However, the fast searching algorithm introduced in [4] assumes an infinite lattice which can not be used as the codebook in real-time audio coding systems. In other words, for a given rate the algorithm can not be used to quantize the input vectors lying outside the truncated lattice region. To resolve this problem, a fast method for quantizing “outliers” $x_o$ is developed and described as follows:

1) Scale down the vector $x_o$ by 2: $x_o = x_o / 2$.
2) Find the nearest $D_8$ point $u$ to $x_o$ and then compute the index vector $j$ of $u$ using the indexing algorithm in [4].
3) Find the codeword $y$ from the index vector $j$ and then compare $y$ with $u$. If $y$ is different from $u$, repeat Steps 1) to 3). Otherwise, compute $w = x_o / 16$. Note that due to the normalization of transform coefficients, a few iterations are required to find the best codeword.
4) Compute $x_o = x_o + w$.
5) Find the nearest $D_8$ point $u$ to $x_o$ and then compute the index vector $j$ of $u$ using the indexing algorithm in [4].
6) Find the codeword $y$ from the index vector $j$ and then compare $y$ with $u$. If $y$ and $u$ are exactly same, $k = j$ and repeat Steps 4) to 6). Otherwise, $k$ is the index of the best codeword to $x_o$ and stop.

It has been observed that the 8-dimensional coefficient vectors have a high concentration of probability around the origin. Therefore, Huffman coding is optionally tried for the quantization indices of HRQ to increase efficiency of quantization when the rate is smaller than 6 bit/coefficient.

LRQ is a 1-bit quantizer and uses spherical codes based on the rotated Gosset lattice $RE_8$ [5] as the codebook. The $RE_8$ lattice is defined as

$$RE_8 = 2 \cdot D_8 \cup \{2 \cdot D_8 + (1, 1, 1, 1, 1, 1, 1, 1)\}$$

and consists of the points falling on concentric spheres of radius $2\sqrt{2}r$ centered at the origin, where $r = 0, 1, 2, \ldots$. The set of points on a sphere constitutes a spherical code and can be used as a quantization codebook. The LRQ codebook of 256 codewords is stored in a structured look-up table so that a fast searching method and a fast indexing method are developed to reduce the computational complexity [3].

5. COMPLEXITY OF THE G.719 CODEC

The computational complexity of the G.719 codec in 16/32-bit fixed-point is estimated by encoding and decoding the source material used for the subjective tests of the G.719 Optimization/Characterization phase. The observed average and worst-case complexity in units of WMOPS [6] are shown in Table 1 for different bitrates. Storage requirements are reported in Table 2.

<table>
<thead>
<tr>
<th>Bitrate (kbps)</th>
<th>Encoder</th>
<th>Decoder</th>
<th>Encoder + Decoder</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>6.663</td>
<td>7.996</td>
<td>13.539</td>
</tr>
<tr>
<td>64</td>
<td>7.912</td>
<td>9.899</td>
<td>17.811</td>
</tr>
<tr>
<td>80</td>
<td>8.303</td>
<td>10.564</td>
<td>18.867</td>
</tr>
<tr>
<td>96</td>
<td>8.555</td>
<td>10.796</td>
<td>19.353</td>
</tr>
<tr>
<td>112</td>
<td>8.787</td>
<td>10.881</td>
<td>19.668</td>
</tr>
<tr>
<td>128</td>
<td>8.980</td>
<td>11.775</td>
<td>20.755</td>
</tr>
</tbody>
</table>

Table 1: Computational complexity of the G.719 codec.

<table>
<thead>
<tr>
<th>Memory type</th>
<th>Encoder</th>
<th>Decoder</th>
<th>Encoder + Decoder</th>
</tr>
</thead>
<tbody>
<tr>
<td>Static RAM</td>
<td>1.0</td>
<td>3.9</td>
<td>4.9</td>
</tr>
<tr>
<td>Scratch RAM</td>
<td>12.2</td>
<td>12.2</td>
<td>24.4</td>
</tr>
<tr>
<td>Data ROM</td>
<td>8.3</td>
<td>8.9</td>
<td>10.7</td>
</tr>
<tr>
<td>Program ROM</td>
<td>1.2</td>
<td>1.2</td>
<td>1.8</td>
</tr>
</tbody>
</table>

1 In units of 16-b kWords 2 In units of 1000’s basic operators

6. SUBJECTIVE PERFORMANCE OF G.719

Subjective tests for the ITU-T G.719 (ex. G.722.1FB) Optimization/Characterization phase were performed from mid February through early April 2008 by independent listening laboratories in American English, French, and Spanish. According to a test plan designed by ITU-T Q7/SG12 experts, the joint candidate codec conducted two experiments as follows:

- Experiment 1: Speech (clean, reverberant, and noisy)
- Experiment 2: Mixed content and music

Mixed content items are representative of advertisement, film trailers, news with jingles, music with announcements, etc., and contain speech, music, and noise. Each experiment used the “triple stimulus/hidden reference/double blind test method” described in ITU-R Recommendation BS.1116-1. A standard MPEG audio codec, LAME MP3 version 3.97 as found on the LAME website as of September 2006 [7] was used as the reference codec in the subjective tests. The ITU-T requirement
was that the G.719 candidate codec at 32, 48, and 64 kbps be proven “Not Worse Than” the reference codec at 40, 56, and 64 kbps, respectively, with a 95% statistical confidence level. In addition, the G.719 candidate codec at 64 kbps was also tested against the G.722.1C codecs at 48 kbps for Experiment 2.

The subjective test results for the G.719 codec are shown in Figures 3-6. Statistical analysis of the results showed that the G.719 codec met all performance requirements specified for the subjective Optimization/Characterization test. For experiment 1 the G.719 codec was better than the reference codec at all bit rates in both languages. For experiment 2 the G.719 codec is better than the reference codec at the lowest bit rate for all the items and at the two other bitrates for most of the items.

An additional subjective listening test for the G.719 codec was conducted later to evaluate the quality of the codec at rates higher than those described in the ITU-T test plan. Because the quality expectation of the codec at these high rates is high, a pre-selection of critical items, for which the quality at the lower bitrate range was most degraded, was conducted prior to testing. The test results are shown in Figure 7. It has been proven that transparency was reached for critical material at 128 kbps.

7. CONCLUSION

This paper has presented the 20 kHz full-band audio coding algorithm and subjective Optimization/Characterization test results of ITU-T Recommendation G.719. The G.719 codec features very high audio quality, extremely low computational complexity, and low algorithmic delay compared to other state-of-the-art audio coding codecs. Subjective test results show that the G.719 codec achieves transparent audio quality at 128 kbps. The main intended applications for this codec are videoconferencing and teleconferencing, as well as Internet streaming.

8. REFERENCES